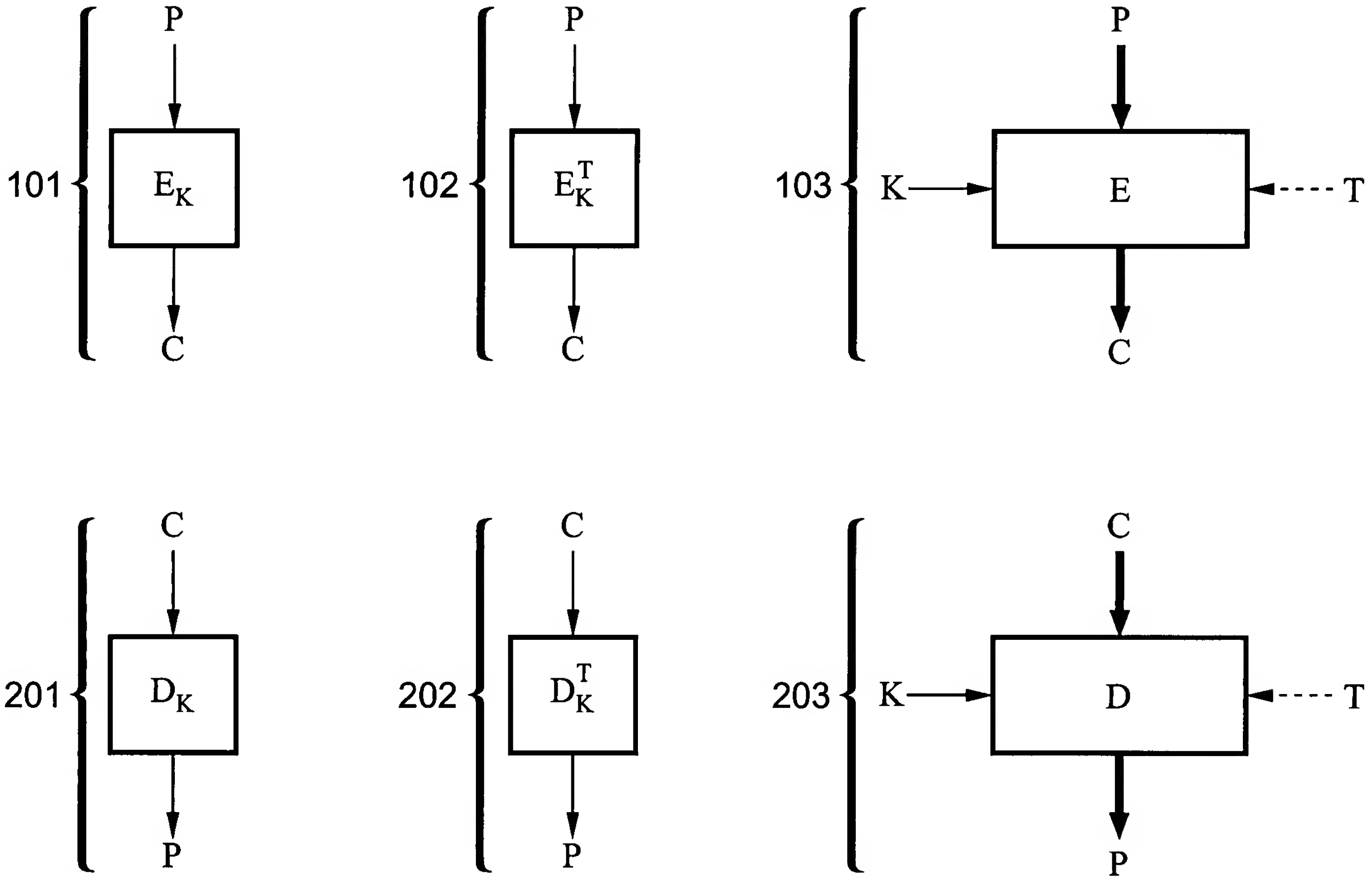


FIG. 1

```
Algorithm $E_K$ (P)                                     // ECB encipher
Partition P into $P_1 \cdots P_m$   (where each $P_i$ is n-bits)
for i ←1 in [1 .. m] do
    $C_i \leftarrow E_K(P_i)$
return $C_1 \cdots C_m$

Algorithm $D_K$ (C)                                     // ECB decipher
Partition C into $C_1 \cdots C_m$   (where each $C_i$ is n-bits)
for i ←1 in [1 .. m] do
    $P_i \leftarrow E_K^{-1}(C_i)$
return $P_1 \cdots P_m$
```

FIG. 2

```
Algorithm $\mathbf{E}_K$ (P)                                    // CBC encipher
Partition P into $P_1 \cdots P_m$     (where each $P_i$ is n-bits)
$C_0 \leftarrow 0^n$
for $i \leftarrow 1$ to m do
    $C_i \leftarrow E_K(C_{i-1} \oplus P_i)$
return $C_1 \cdots C_m$

Algorithm $\mathbf{D}_K$ (C)                                    // CBC decipher
Partition C into $C_1 \cdots C_m$     (where each $C_i$ is n-bits)
$C_0 \leftarrow 0^n$
for $i \in [1..m]$ do
    $P_i \leftarrow E_K^{-1}(C_{i-1}) \oplus C_{i-1}$
return $P_1 \cdots P_m$
```

FIG. 3

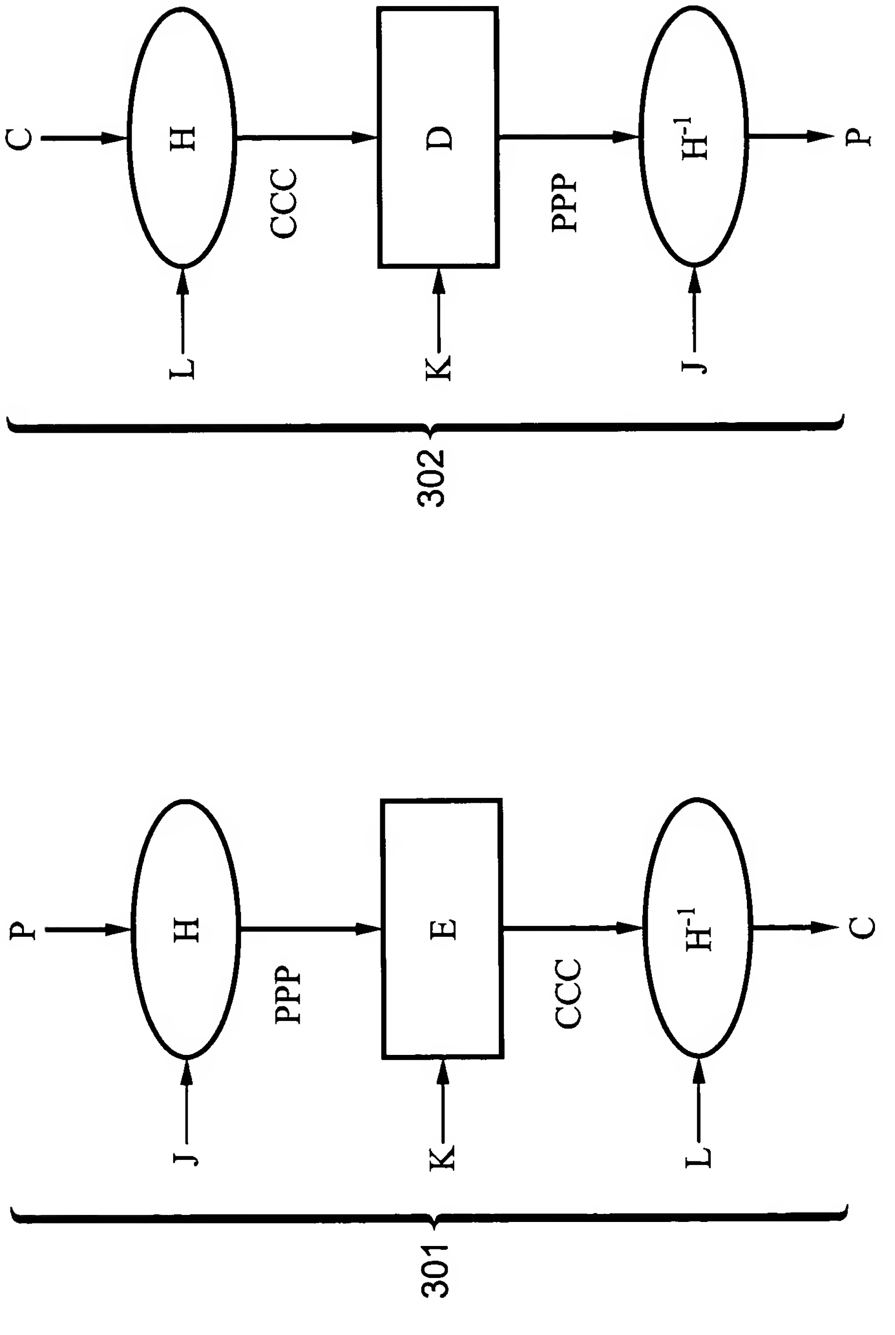


FIG. 4

```
Algorithm double (S)      // assumes $|S| = 128$ and $P_{128}(\mathbf{x}) = \mathbf{x}^{128}+\mathbf{x}^7+\mathbf{x}^2+\mathbf{x}+1$

100  if firstbit (S) = 0 then return $S \ll 1$                         // left shift
101                          else return $(S \ll 1) \oplus 0^{120}10000111$   // left shift & xor
```

FIG. 5

```
**Algorithm** $\mathbf{E}_K(P)$                                   // *CMC encipher*
100 Partition P into $P_1 \cdots P_m$    (where each $P_i$ is n-bits)

110 $PPP_0 \leftarrow 0^n$                                        // *Encipher*
111 **for** $i \leftarrow 1$ **to** m **do**
112     $PP_i \leftarrow P_i \oplus PPP_{i-1}$
113     $PPP_i \leftarrow E_K(PP_i)$

120 $M \leftarrow 2(PPP_1 \oplus PPP_m)$                            // *Mask*
121 **for** $i \in [1..m]$ **do** $CCC_i \leftarrow PPP_{m+1-i} \oplus M$

130 $CCC_0 \leftarrow 0^n$                                        // *Decipher*
131 **for** $i \leftarrow 1$ **to** m **do**
132     $CC_i \leftarrow E_K(CCC_i)$
133     $C_i \leftarrow CC_i \oplus CCC_{i-1}$

140 **return** $C_1 \cdots C_m$
```

FIG. 6

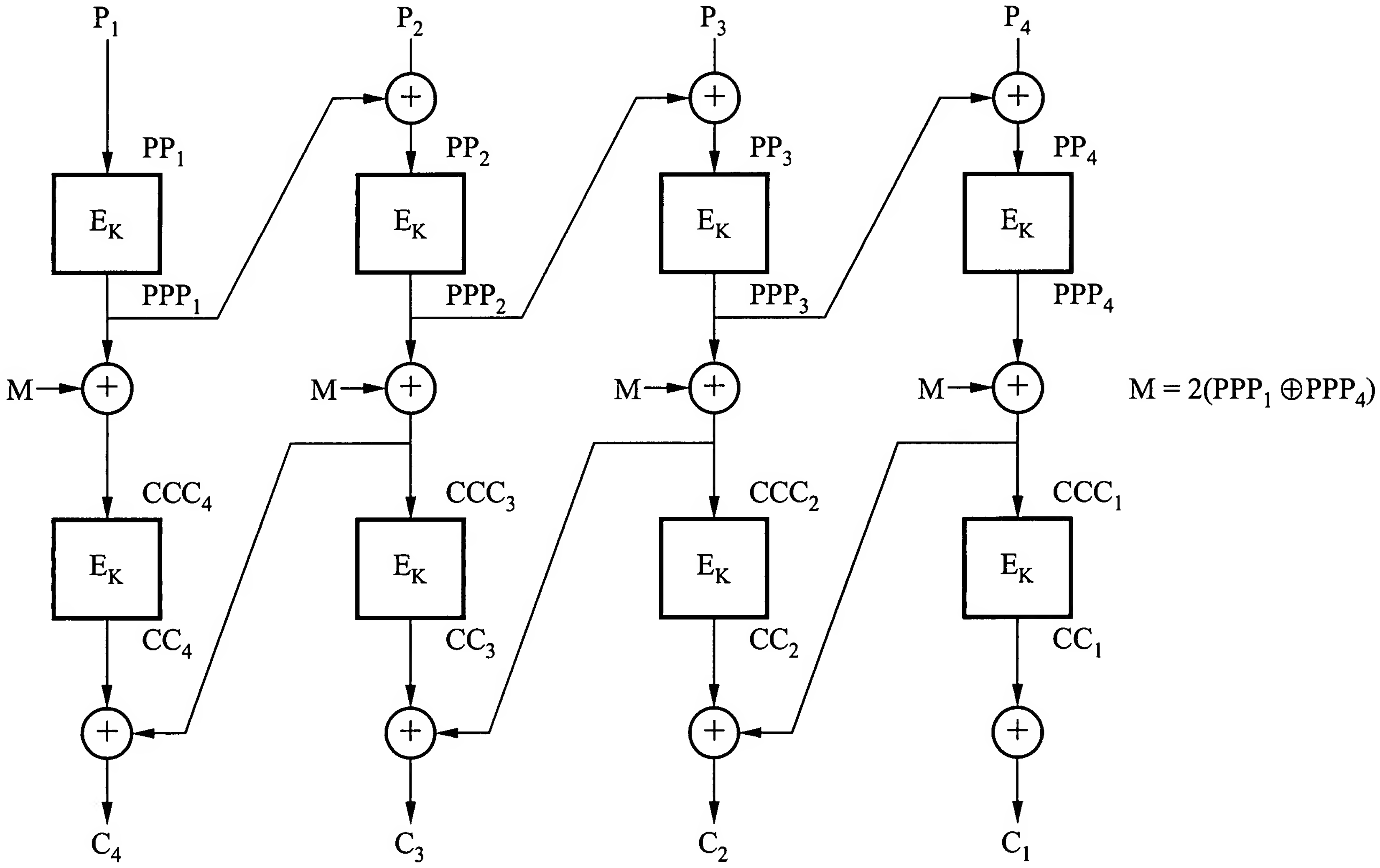

P_1
P_2
P_3
P_4
PP_1
PP_2
PP_3
PP_4
E_K
PPP_1
PPP_2
PPP_3
PPP_4
M
$M = 2(PPP_1 \oplus PPP_4)$
CCC_4
CCC_3
CCC_2
CCC_1
CC_4
CC_3
CC_2
CC_1
C_4
C_3
C_2
C_1

FIG. 7

```
**Algorithm** $\mathbf{D}_K$ (C)                                   // CMC *decipher*
Partition C into $C_1 \cdots C_m$   (where each $C_i$ is n-bits)

$CCC_0 \leftarrow 0^n$                                             // *Encipher*
**for** $i \leftarrow 1$ **to** m **do**
    $CC_i \leftarrow C_i \oplus CCC_{i-1}$
    $CCC_i \leftarrow E_K^{-1}(CC_i)$

$M \leftarrow 2(CCC_1 \oplus CCC_m)$                               // *Mask*
**for** $i \in [1 .. m]$ **do** $PPP_i \leftarrow CCC_{m+1-i} \oplus M$

$PPP_0 \leftarrow 0^n$                                             // *Decipher*
**for** $i \leftarrow 1$ **to** m **do**
    $PP_i \leftarrow E_K^{-1}(PPP_i)$
    $P_i \leftarrow PP_i \oplus PPP_{i-1}$

**return** $P_1 \cdots P_m$
```

FIG. 8

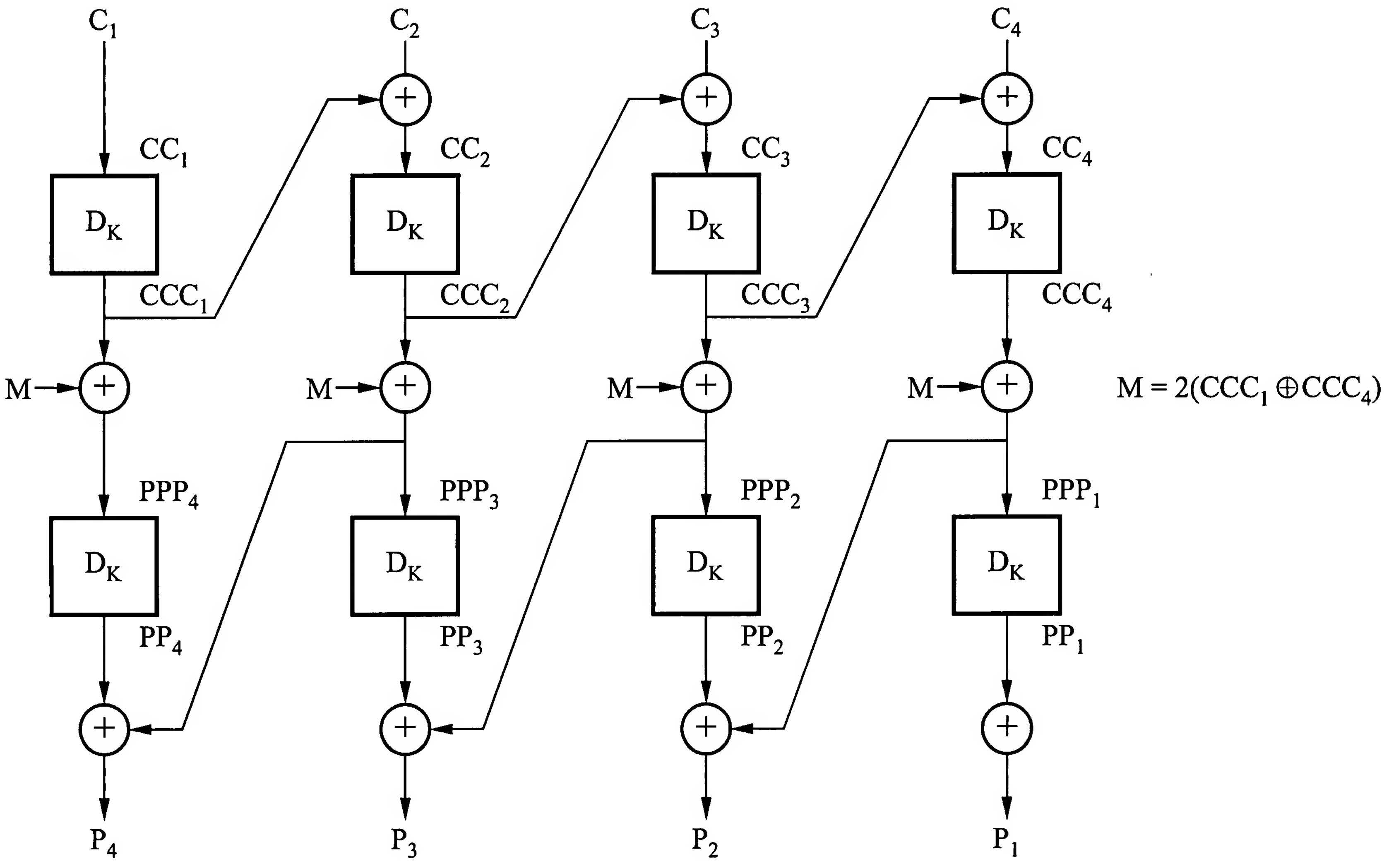

C_1
C_2
C_3
C_4
CC_1
CC_2
CC_3
CC_4
D_K
CCC_1
CCC_2
CCC_3
CCC_4
M
$M = 2(CCC_1 \oplus CCC_4)$
PPP_4
PPP_3
PPP_2
PPP_1
PP_4
PP_3
PP_2
PP_1
P_4
P_3
P_2
P_1

FIG. 9

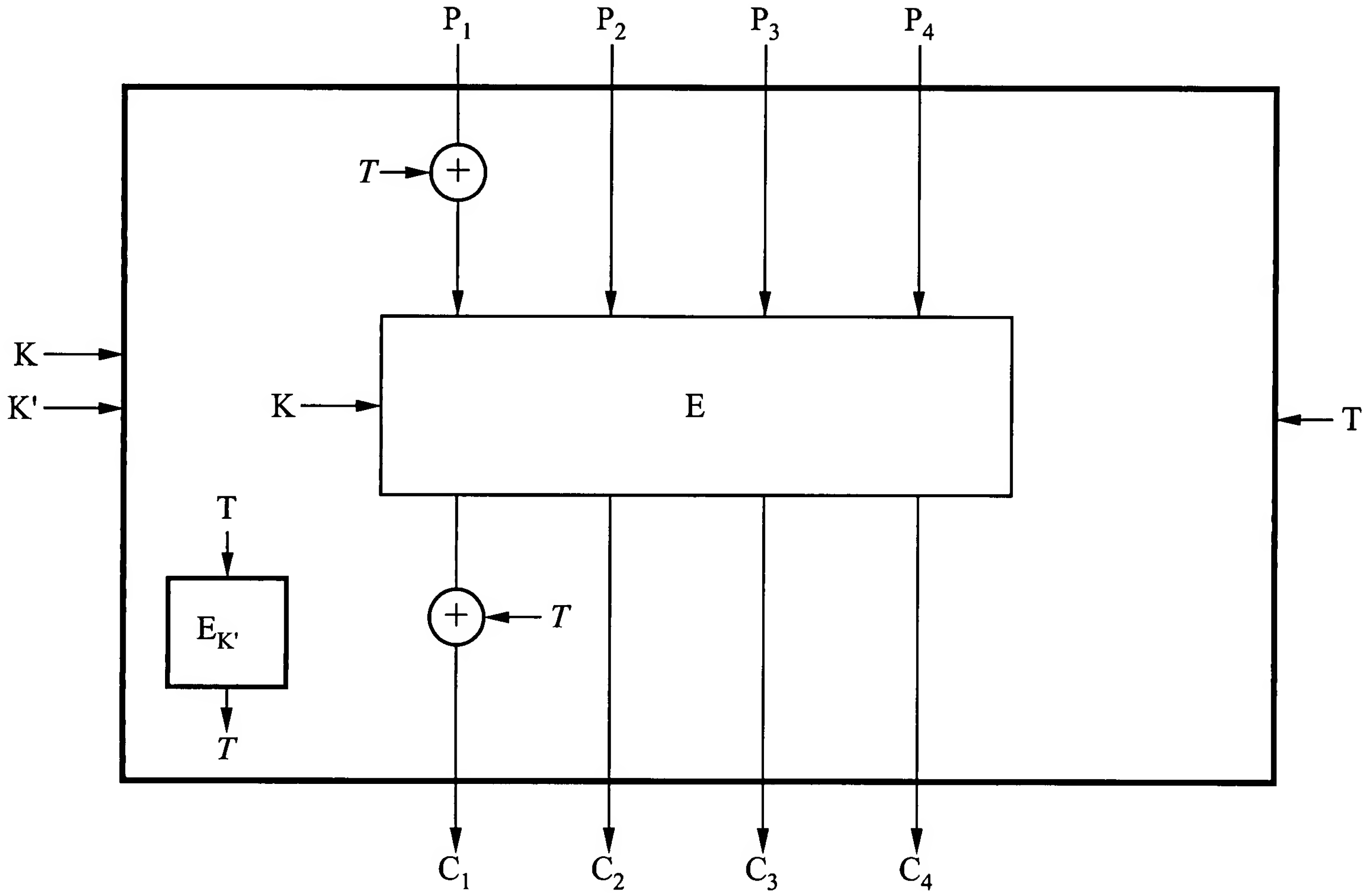


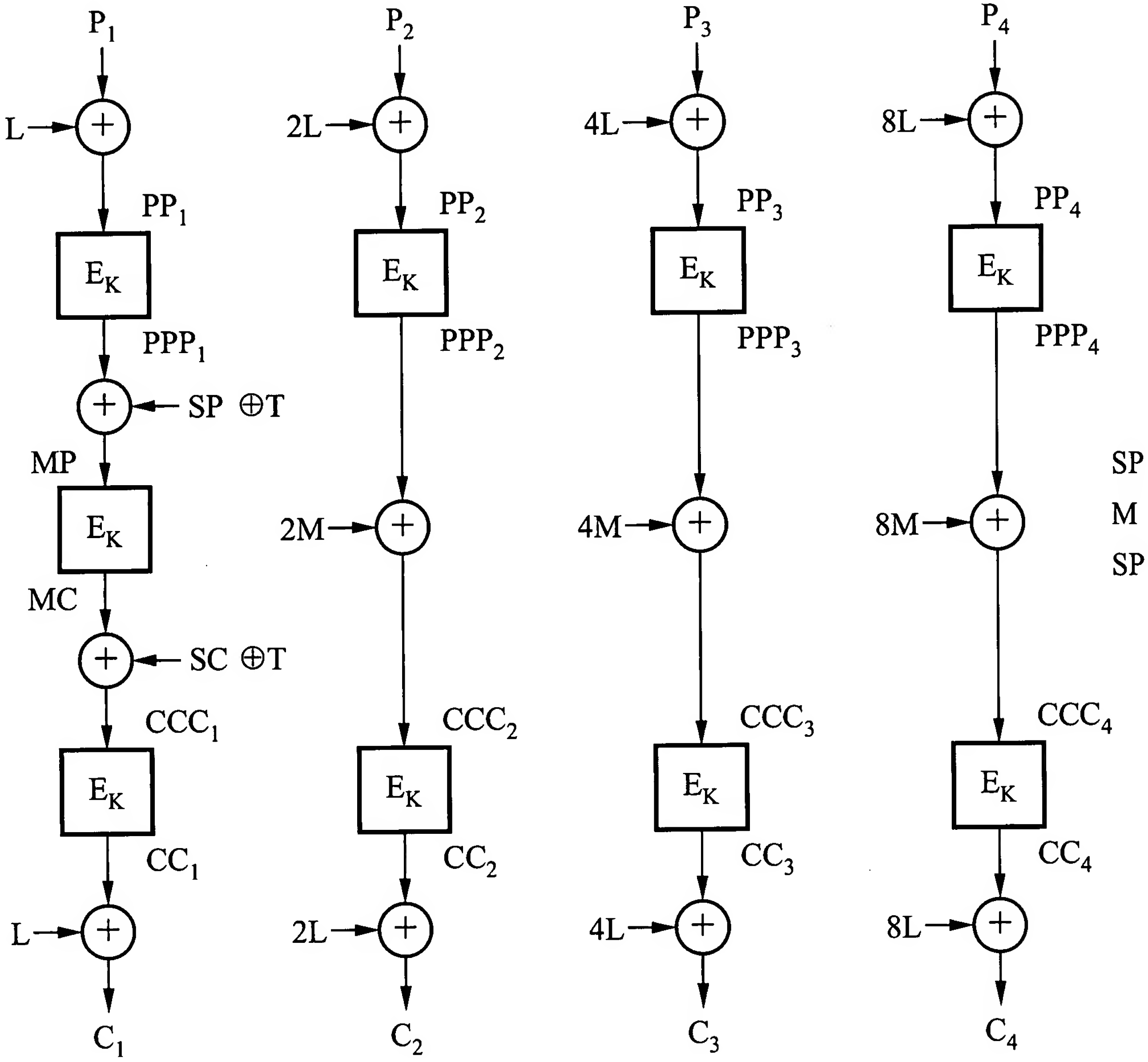
FIG. 10

FIG. 11

```
**Algorithm** $\mathbf{E}_{K\ K'}$ (T, P)                                  // *tweakable CMC encipher*
100  Partition P into $P_1 \cdots P_m$    (where each $P_i$ is n-bits)
101  $\boldsymbol{T} \leftarrow E_{K'}(T)$

110  $PPP_0 \leftarrow \boldsymbol{T}$                                      // *Encipher*
111  **for** $i \leftarrow 1$ **to** m **do**
112      $PP_i \leftarrow P_i \oplus PPP_{i-1}$
113      $PPP_i \leftarrow E_K(PP_i)$

120  $M \leftarrow 2(PPP_1 \oplus PPP_m)$                                  // *Mask*
121  **for** $i \in [1 .. m]$ **do** $CCC_i \leftarrow PPP_{m+1-i} \oplus M$

130  $CCC_0 \leftarrow 0^n$                                             // *Decipher*
131  **for** $i \leftarrow 1$ **to** m **do**
132      $CC_i \leftarrow E_K(CCC_i)$
133      $C_i \leftarrow CC_i \oplus CCC_{i-1}$
134  $C_1 \leftarrow C_1 \oplus \boldsymbol{T}$

140  **return** $C_1 \cdots C_m$
```

```
**Algorithm** $\mathbf{E}_K^T(P_1 \cdots P_m)$                                  // *EME encipher*
100  $L \leftarrow 2E_K(0^n)$

101  **for** $i \leftarrow 1 \in [1 .. m]$ **do**                               // *Encipher*
102        $PP_i \leftarrow 2^{i-1} L \oplus P_i$
103        $PPP_i \leftarrow E_K(PP_i)$

110  $SP \leftarrow PPP_2 \oplus \cdots \oplus PPP_m$                             // *Mask*
111  $MP \leftarrow PPP_1 \oplus SP \oplus T$
112  $MC \leftarrow E_K(MP)$
113  $M \leftarrow MP \oplus MC$
114  **for** $i \in [1 .. m]$ **do** $CCC_i \leftarrow PPP_i \oplus 2^{i-1} M$
115  $SC \leftarrow CCC_2 \oplus \cdots \oplus CCC_m$
116  $CCC_1 \leftarrow MC \oplus SC \oplus T$

120  **for** $i \in [1 .. m]$ **do**                                           // *Decipher*
121        $CC_i \leftarrow E_K(CCC_i)$
122        $C_i \leftarrow CC_i \oplus 2^{i-2} L$

130  **return** $C_1 \cdots C_m$
```

FIG. 12



$SP = PPP_2 \oplus PPP_3 \oplus PPP_4$

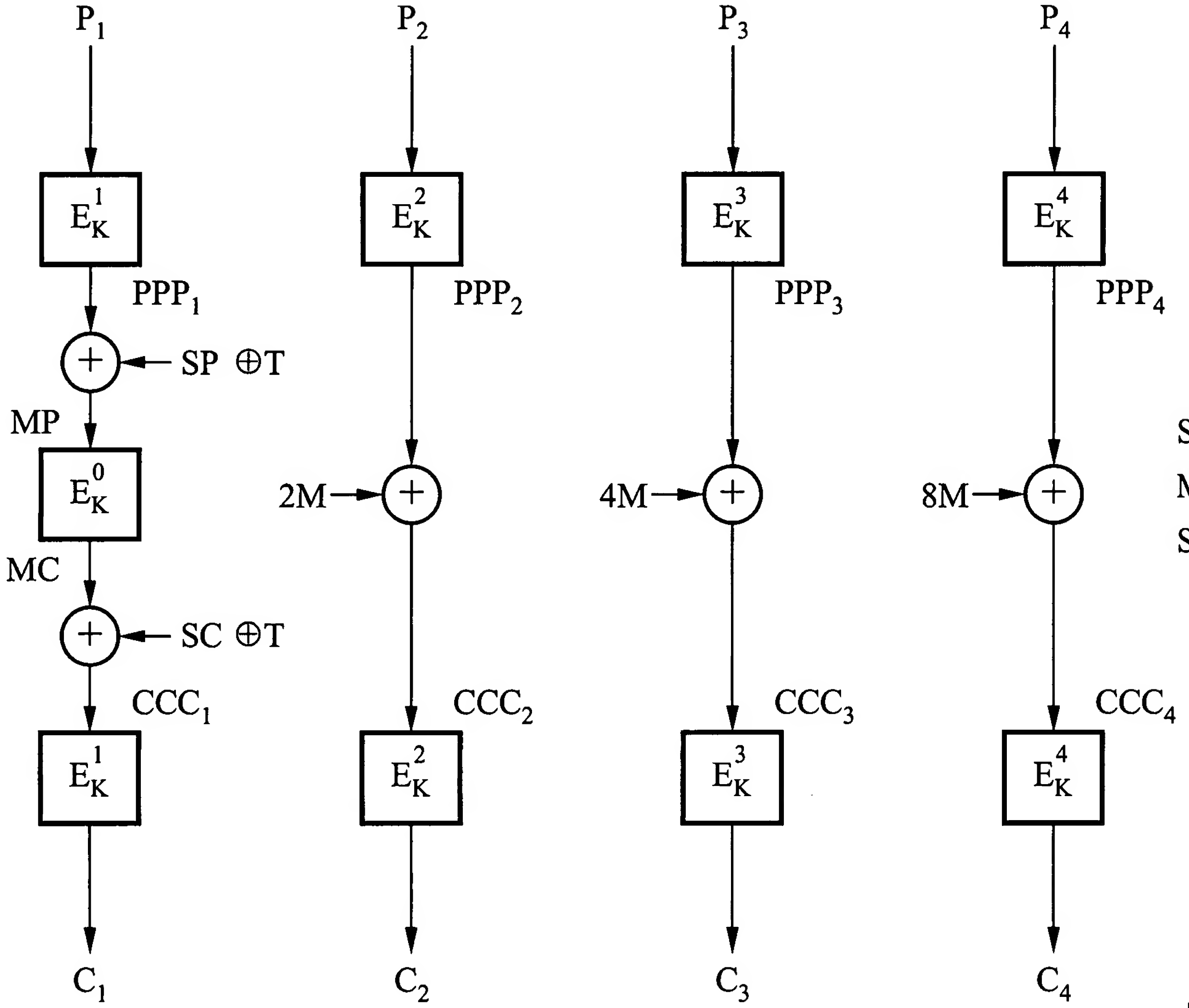
$M \ = MP \oplus MC$

$SP = CCC_2 \oplus CCC_3 \oplus CCC_4$

FIG. 13

```
Algorithm D $_K^T(C_1 \cdots C_m)$                              // EME decipher
100  $L \leftarrow 2E_K(0^n)$

101  for $i \leftarrow 1 \in [1 .. m]$ do                          // Encipher
102      $PP_i \leftarrow 2^{i-1} L \oplus C_i$
103      $PPP_i \leftarrow D_K(CC_i)$

110  $SC \leftarrow CCC_2 \oplus \cdots \oplus CCC_m$               // Mask
111  $MC \leftarrow CCC_1 \oplus SC \oplus T$
112  $MP \leftarrow D_K(MC)$
113  $M \leftarrow MC \oplus MP$
114  for $i \in [1 .. m]$ do $PPP_i \leftarrow CCC_i \oplus 2^{i-1} M$
115  $SP \leftarrow PPP_2 \oplus \cdots \oplus PPP_m$
116  $PPP_1 \leftarrow MP \oplus SP \oplus T$

120  for $i \in [1 .. m]$ do                                        // Decipher
121      $PP_i \leftarrow D_K(PPP_i)$
122      $P_i \leftarrow PP_i \oplus 2^{i-2} L$

130  return $P_1 \cdots P_m$
```

FIG. 14



$$SP = PPP_2 \oplus PPP_3 \oplus PPP_4$$
$$M = MP \oplus MC$$
$$SP = CCC_2 \oplus CCC_3 \oplus CCC_4$$

FIG. 15